



IMASの概要とLHDデータの IMAS-IDS化の初期的取り組みの紹介

草場 穫

核融合科学研究所 学際連携センター 特任助教

日時 2026年2月20日

目次

1. IMASの概要

→p3~9

2. LHDデータのIMAS-IDS化の初期的取り組みの紹介

→p10~27

1. IMASの概要

IMASとは？

概要：IMAS (Integrated Modelling & Analysis Suite) は、ITERを中心に開発されてきた統合モデリング・解析のためのソフトウェアフレームワークであり、プラズマシミュレーションおよび実験データ解析を共通のデータモデル (IMAS-IDS) に基づいて標準化・効率化することを目的としている。

- IMAS (Integrated Modelling & Analysis Suite) → **枠組み全体**

- ITER を中心に国際的に整備されている、核融合分野の統一データ基盤

- 装置ごとに異なる診断データを共通データモデルで表現

- IDS (Integrated Data Structure) → **共通データモデル (DDに従うデータを入れる箱)**

- DD (Data Dictionary) に従って実際に使うデータコンテナ

- 1つの IDS = 特定の物理概念や装置ごとにデータをまとめた箱 (ノードを最小単位とする木構造になっている)

- DD (Data Dictionary) とは? → **設計図**

- 共通データモデルの設計図であり、XML 形式で定義されたすべての IDS の構造仕様を提供

- どんなノードがあるか、型は何か、必須か任意かが書かれている

公開状況：2025年12月上旬にIMASのデータモデルやAPIを含む基盤ソフトウェアは、GitHub上でオープンソースとして公開された (プレスリリース: <https://www.iter.org/node/20687/release-imas-infrastructure-and-physics-models-open-source>, GitHub: <https://github.com/iterorganization>) 。

IMASとは?

• どのようなIDSがあるか? 各IDSがどのようなノードを持つかは、IMAS Data Dictionary documentationで確認できる。

• IMAS-IDSを使う場合は、このドキュメントを読み込んで、装置で測定された各データをどのIDSのどのノードに割り当てるかを定める必要がある(対応表の作成)。

• IDS reference

- α amns_data
- α b_field_non_axisymmetric
- α balance_of_plant
- α barometry
- α bolometer
- α breeding_blanket
- α bremsstrahlung_visible
- α calorimetry
- α camera_ir
- α camera_visible
- α camera_x_rays
- α charge_exchange
- α coils_non_axisymmetric
- α controllers
- ★ core_instant_changes
- ★ core_profiles

利用可能なIDS一覧

- α lh_antennas
- ★ magnetics
- α mhd
- α mhd_linear
- α mse
- ★ nbi
- α neutron_diagnostic
- α ntms
- α operational_instrumentation
- α pellets
- ★ pf_active
- α pf_passive
- α pf_plasma
- α plasma_initiation
- α plasma_profiles
- α plasma_sources

★ core_profiles 特定のIDSの説明

Core plasma profiles + Expand all
- Collapse all

- Maximum occurrences (MDS+ backend only): 15

! New in version 3.1.0: lifecycle status active

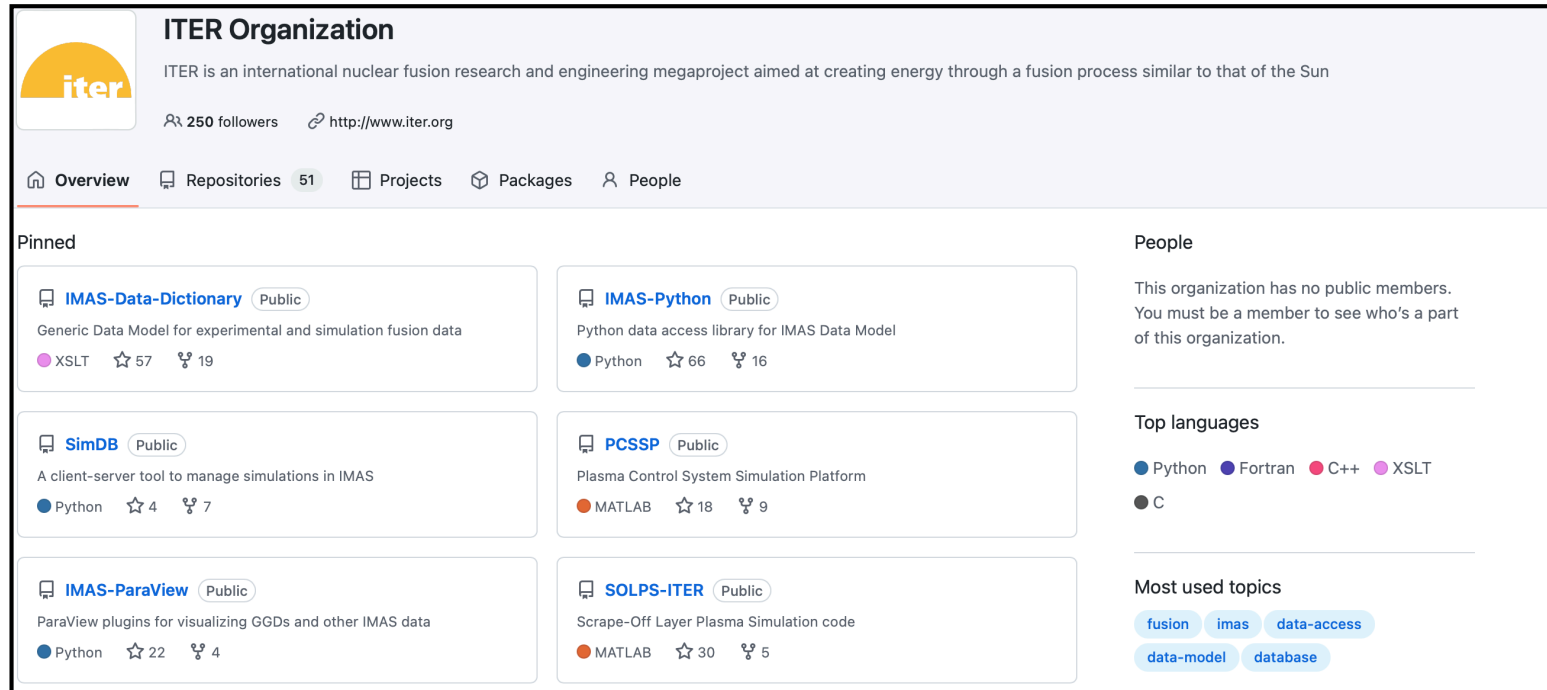
! Changed in version 4.2.0.

- > ids_properties structure
- > profiles_1d(itime) (dynamic) AoS
Core plasma radial profiles for various time slices
- > profiles_2d(itime) (dynamic) AoS
Core plasma quantities in a poloidal cross section, for various [...]
- > global_quantities structure
Various global quantities derived from the profiles

core_profiles IDSの各ノードの説明

現在公開されているIMAS関連コードの概要

- ITER OrganizationのGitHub（公式公開リポジトリ群）では、以下のようなリポジトリを公開中



*現在(2026/2/20)、51のリポジトリが公開されているが、主要なものは以下となる。

1. コア/データモデル関連

- **IMAS-Core:** IMASのAccess Layerのコアライブラリ。IDSを読み書きするための低レベルライブラリ。
- **IMAS-Data-Dictionary:** IMASデータモデル（IDS）のXMLによるデータディクショナリ定義（データの名前、構造化の説明）。

2. Python/可視化

- **IMAS-Python:** IMASデータモデル（IDS）用のPythonライブラリ。IDSの生成、読み書き、加工をPythonから行うためのツール。基本的な機能はIMAS-Coreに依存せず、IMAS-Coreが利用可能な場合にはHDF5バックエンドや高度なデータアクセス機能も利用できる。
- **IMAS-ParaView:** ParaViewとの連携プラグイン。IMASデータの可視化ツール（インタラクティブな3D図が書ける。環境構築難）。

現在公開されているIMAS関連コードの概要

コードのライセンスについて

- IMAS-Core、IMAS-Python、IMAS-ParaView は LGPL-3.0 license のもとで公開されている。
 - IMAS-Data-Dictionary はコンテンツの種類ごとにライセンスが分けられており、データスキーマ（IDS 定義などの標準仕様部分）は CC-BY-ND 4.0、付随するソフトウェア部分は LGPL-3.0 が適用されている（IMAS-Data-Dictionary リポジトリの Legal セクションに記載）。
- 基本的にIMAS関連コードは LGPL-3.0 license で公開されており、特別な契約無しに自由に使用することができる（商用利用も可）。
- ITER 内部のデータ資産（非公開データ）、未公開ツール、および公式運用環境へのアクセスしたい場合のみ要ITERアカウント。

結論：NIFS内において IMAS 関連コードを自由に運用することが可能である。元の IMAS コードをリンクして利用する形であれば、独自の拡張コードを作成・運用することも可能であり、商用利用（企業との共同研究等）も可能である。

→ヘリカル拡張コードや Data Dictionary 拡張をITER公式（IMAS公式）として発表・登録したい場合は、ITERアカウントが必要（+他のヘリカル研究組織との合意）。

IDSの保存形式・IMAS-CoreとIMAS-Pythonの違い

- Access Layer を担う IMAS-Core は、ITER 公式データベース運用を想定し、HDF5 形式を用いたデータ保存・アクセス機構を提供している。一方、Python ライブラリである IMAS-Python（高水準API）は、IMAS-Core に依存しない利用を可能とするため、netCDF4 を用いた保存バックエンドを標準で備えている。IMAS-Python から HDF5 バックエンドを利用する場合には、IMAS-Core の導入が必要となる。さらに IMAS-Core を導入することで、slicing に代表される高度な部分読み出し機能や、MDSplus を用いたデータ保存・アクセスも可能となる。
- 軽量な解析環境において IMAS を用いた解析・共有を行う場合には、IMAS-Python のみで十分である。一方、本格的に IMAS-IDS に基づく大規模データベースを運用する場合には、IMAS-Pythonに加えてIMAS-Core も導入することが望ましい。

IMAS-IDS活用のメリット

1. データの意味が固定される。
→量名・単位・次元・時間軸が Data Dictionary で厳密に定義されるため、解釈ズレが起きにくい
2. HDF5系で保存されるので、部分読み出しに強い。
3. データが装置横断的に使えるようになる。
4. ショットごとに物理量を体系的にまとめることができる。
5. IMAS-IDS I/Oを前提とした、既存・将来のコード資産にそのまま投げることができる。

→以上より、IMAS-IDS形式でデータを保存しておく、共同研究・外部連携が格段に容易になると想定される。一方で、IMAS-IDSは基本的にトカマク中心に開発されているので、ヘリカル特有の量に関してはコードの拡張やIDSでの保存はそもそも行わないなど対策が必要である。

・ NIFS内における想定されるIMAS-IDS活用法

1. LHD解析データをIMAS-IDSに変換し、保存・公開する。

→大量の診断名があるので、主要なもの一部から変換を試みる（初期的取り組みを後述する）。

2. CHDデータ解析システムでの活用。

→IMAS-IDSは外に出すための整理・標準化されたデータと考えるのが妥当であり、データ解析システム内のデータ形式は既存のEG形式で運用することを想定している。よって、外部アダプター的なIMAS-IDS形式への変換・保存用コードを作れば十分であり、データ解析システムの開発のボトルネックにはならない。1で作成したコードを多くの部分で流用できると想定している。

IMAS-IDS活用に必要なこと

1. CHD / LHD で測定された各診断データについて、IMAS-IDS 上での格納先 (IDS / ノード / 物理的意味) を定義する対応表を整備する (最重要)。

→各EG形式診断データが持つ物理的意味を、IMAS-IDS データモデルが定義する物理概念へと翻訳する作業であり、単なるフォーマット変換ではないことに注意。これは、NIFS所内での合意に基づき設計する必要がある。座標変換(必要な場合)や外れ値・欠損値の扱いもここに含まれる。

LHD 診断名 = thomsonに
おける例：

意味	LHD項目	LHD単位	IMAS項目	IMAS単位	変換・処理	IMAS項目の定義
時間	Time	ms	core_profiles.time[:]	s	$t_s = \text{Time} \times 1e-3$	IDS全体で共通に用いる基準時刻配列
時間	Time	ms	profiles_1d[it].time	s	$t_s = \text{Time} \times 1e-3$	各1次元プロファイルが対応する時刻
動径位置	R	mm	profiles_1d[it].grid.rho_tor_norm[:]	-	$\rho = (R \times 1e-3) / a$ 動径方向に整列	正規化トロイダル磁束半径($\theta=1$ の動径座標)

2. 1の対応表に基づいたEG形式→IMAS-IDS形式変換コードの作成

→この部分は外注することができる。一方対応表のデザインは専門知識が必要なので外注は難しい。

3. ヘリカル・装置特有の量を保存できるようにコードを拡張する (作業として重い)

→Data Dictionaryからいじる必要があるか？ ヘリカルをトカマクに座標変換する外部ライブラリのようなものを作成するか (既存のものは使えるか) ? そもそもIDSとして保存するメリットがあるか？

など診断名ごとに要議論。とりあえずは、3が必要なく対応できる主要診断名を中心に変換コードを作成する予定。

2. LHDデータのIMAS-IDS化の初期的取り組みの紹介

変換対象診断データと実行方法

- 今回IDS変換の対象とした診断名は以下の通り

→LHD解析データリポジトリの以下診断名に対応するEG形式テキストファイルをIDSに変換して保存する

項目	shotinfo	wp	THOMSON
データの役割	実験基本情報	全体的なプラズマ量	プラズマ内部 プロフィール
時間依存性	あり(1点)	あり	あり
空間依存性	なし	なし	あり
対応する IMAS IDS	summary	summary	core_profiles

- 実行方法

IMAS-Python (IMAS-Core非依存) を使って、上記3つの診断名をIDSに変換し、ショットごとに統合して netcdf4ファイルとして保存するコードを作成する。また、matplotlibを使った簡単な可視化も行う。

まず対象診断名の選択、対応表のデザインを行い、その表に基づいて変換コードを作成する。

診断名=thomsonの対応表 (IDS=core_profiles)

・ 時間・空間軸の設定

意味	LHD項目	LHD単位	IMAS項目	IMAS単位	変換・処理	IMAS項目の定義
時間	Time	ms	core_profiles.time[:]	s	$t_s = \text{Time} \times 1e-3$	IDS全体で共通に用いる基準時刻配列
時間	Time	ms	profiles_1d[it].time	s	$t_s = \text{Time} \times 1e-3$	各1次元プロファイルが対応する時刻
動径位置	R	mm	profiles_1d[it].grid.rho_tor_norm[:]	-	$\rho = (R \times 1e-3) / a$ 動径方向に整列	正規化トロイダル磁束半径($\theta=1$ の動径座標)

・ 電子温度

意味	LHD項目	LHD単位	IMAS項目	IMAS単位	変換・処理	IMAS項目の定義
電子温度	Te	eV	profiles_1d[it].electrons.temperature[:]	eV	$Te \geq 9.9e5 \rightarrow \text{NaN}$ 動径方向に整列	電子の局所熱平衡温度プロファイル
温度誤差	dTe	eV	profiles_1d[it].electrons.temperature_error_upper[:]	eV	$dTe \geq 9.9e5 \rightarrow \text{NaN}$ 動径方向に整列	電子温度の上側不確かさ
温度誤差	dTe	eV	profiles_1d[it].electrons.temperature_error_lower[:]	eV	$dTe \geq 9.9e5 \rightarrow \text{NaN}$ 動径方向に整列	電子温度の下側不確かさ

診断名=thomsonの対応表 (IDS=core_profiles)

・電子密度

意味	LHD項目	LHD単位	IMAS項目	IMAS単位	変換・処理	IMAS項目の定義
電子密度	n_e	10^{16} m^{-3}	profiles_1d[it].electrons.density_thermal[:]	m^{-3}	$n_e \times 1e16$ $n_e \geq 9.9e5 \rightarrow \text{NaN}$	熱電子成分の数密度プロファイル
密度誤差	dn_e	10^{16} m^{-3}	profiles_1d[it].electrons.density_thermal_error_upper[:]	m^{-3}	$dn_e \times 1e16$ $dn_e \geq 9.9e5 \rightarrow \text{NaN}$ 動径方向に整列	電子密度の上側不確かさ
密度誤差	dn_e	10^{16} m^{-3}	profiles_1d[it].electrons.density_thermal_error_lower[:]	m^{-3}	$dn_e \times 1e16$ $dn_e \geq 9.9e5 \rightarrow \text{NaN}$ 動径方向に整列	電子密度の下側不確かさ

・コメント

情報源	IMAS項目	内容	IMAS項目の定義
LHDコメント+実装説明	ids_properties.comment	出典、単位、近似条件の記述	IDS に関する補足情報

診断名=thomsonの対応表 (IDS=core_profiles)

- 変換後のcore profiles IDSの木構造のイメージ

特定の時刻で切り出したプロファイル例

```
profiles_1d
├── profiles_1d[0]/grid
│   └── rho_tor_norm: array([0.4961, 0.5014, 0.5068, ..., 0.9951, 0.9975, 1.    ], shape=(139,))
├── profiles_1d[0]/electrons
│   ├── temperature: array([ 27.,  34.,  10., ...,  64., 108.,  38.], shape=(139,))
│   ├── temperature_error_upper: array([ 134., 49500., 49500., ..., 49500., 49500., 49500.], shape=(139,))
│   ├── temperature_error_lower: array([ 134., 49500., 49500., ..., 49500., 49500., 49500.], shape=(139,))
│   ├── density_thermal: array([ 9.1e+17, -1.5e+17, 2.8e+17, ..., 1.6e+17, 9.1e+17, 3.8e+17], shape=(139,))
│   ├── density_thermal_error_upper:
│   │   array([ 7.00e+17, -6.70e+17, 7.70e+17, ..., 5.50e+17, 1.06e+18, 7.00e+17], shape=(139,))
│   └── density_thermal_error_lower:
│       array([ 7.00e+17, -6.70e+17, 7.70e+17, ..., 5.50e+17, 1.06e+18, 7.00e+17], shape=(139,))
└── time: 0.016953
```

メタ情報記録部分(ids_properties)

```
ids_properties
├── comment:
│   'LHD Thomson 172000; units: Te[eV], ne[m^-3]; grid: rho_tor_norm ~ r/a (geometric normalization); errors are symmetric (upper=lower).'
```

診断名=shotinfoの対応表 (IDS=summary)

・日時情報

意味	LHD項目	LHD単位	IMAS項目	IMAS単位	変換・処理	IMAS項目の定義
年	year	-	summary.ids_properties.comment	-	文字列として記録	IDS に関する補足情報
月	month	-	summary.ids_properties.comment	-	同上	同上
日	day	-	summary.ids_properties.comment	-	同上	同上
時	hour	-	summary.ids_properties.comment	-	同上	同上
分	minute	-	summary.ids_properties.comment	-	同上	同上
日時文字列	datestr	-	summary.ids_properties.comment	-	同上	同上

・ショット番号

放電番号	ShotNo	-	summary.pulse	-	int に変換	実験放電の識別番号
------	--------	---	---------------	---	---------	-----------

診断名=shotinfoの対応表 (IDS=summary)

・ 磁場 ・ 装置パラメータ

意味	LHD項目	LHD単位	IMAS項目	IMAS単位	変換・処理	IMAS項目の定義
磁場	B	T	summary.global_quantities.b0.value	T	そのまま	基準主半径におけるトロイダル磁場
真空磁気軸半径	Rax_vac	m	summary.global_quantities.r0.value	m	そのまま	真空磁場配位で定義される主半径
ヘリカルピッチ	gamma	-	summary.ids_properties.comment	-	文字列として記録	磁場構成に関する補足情報
磁場比	Bq	%	summary.ids_properties.comment	%	文字列として記録	磁場構成に関する補足情報

ヘリカルピッチと磁場比に対応する項目がないため、コメントに記載

変換後のsummary IDSのメタ情報記録部分のイメージ

```
ids_properties
├── comment:
│   'LHD shot=172000; shotinfo: cycle=23.0, datestr=20211105, YMD=2021-11-05 18:16, Gamma=1.2538, Bq=100.0%; wp
│   integration: start=2.500s, duration=6.831s; wp diagnostics: Wp[kJ], <beta-dia>[%], <beta-vmec>[%].'
```

診断名=wpの対応表 (IDS=summary)

・時間軸の設定

意味	LHD項目	LHD単位	IMAS項目	IMAS単位	変換・処理	IMAS項目の定義
時刻	Time	s	summary.time[:]	s	そのまま	summary における時間基準軸

・プラズマエネルギー

意味	LHD項目	LHD単位	IMAS項目	IMAS単位	変換・処理	IMAS項目の定義
プラズマ内部エネルギー	Wp	kJ	summary.global_quantities.energy_diamagnetic.value	J	Wp × 1e3	反磁性法により評価されたプラズマエネルギー

・ベータ値

意味	LHD項目	LHD単位	IMAS項目	IMAS単位	変換・処理	IMAS項目の定義
反磁性ベータ	beta-dia	%	summary.global_quantities.beta_tor.value	-	% → /100	反磁性信号に基づくトロイダルベータ
VMECベータ	beta-vmec	%	summary.global_quantities.beta_tor_mhd.value	-	% → /100	MHD 平衡再構成に基づくトロイダルベータ

・コメント

情報源	IMAS項目	内容	IMAS項目の定義
LHDコメント	summary.ids_properties.comment	計算コード、仮定、定義式	IDS に関する補足情報

診断名=wpの対応表 (IDS=summary)

変換後のsummary IDSのwp情報格納部分のイメージ

```
global_quantities
├── global_quantities/beta_tor
│   ├── value: array([      nan,          nan,          nan, ...,  2.8089e-07,  6.3932e-08, -0.0000e+00])
│   └── source: 'LHD wp:<beta-dia> (%) -> beta_tor'
├── global_quantities/beta_tor_mhd
│   ├── value: array([      nan,          nan,          nan, ...,  2.2563e-07,  5.1353e-08,  0.0000e+00])
│   └── source: 'LHD wp:<beta-vmec> (%) -> beta_tor_mhd'
├── global_quantities/energy_diamagnetic
│   ├── value: array([      nan,          nan,          nan, ...,  3.4037,  0.7747, -0.    ])
│   └── source: 'LHD wp:Wp (kJ) converted to J'
├── global_quantities/r0
│   ├── value: 3.9
│   └── source: 'LHD shotinfo:Rax_vac'
└── global_quantities/b0
    ├── value: array([-1., -1., -1., ..., -1., -1., -1.])
    └── source: 'LHD shotinfo:B'
```

各IDSをショット番号ごとにまとめて保存

対応するコード

```
# -----  
# Validate and write NetCDF bundle  
# -----  
smry.validate()  
  
with imas.DBEntry(BUNDLE_PATH, "w") as db:  
    db.put(cp)      # core_profiles (Thomson)  
    db.put(smry)   # summary (shotinfo + wp)  
  
print(f"NetCDF bundle saved: {BUNDLE_PATH}")
```



ショットごとに保存されたnetcdf4ファイル

名前	変更日	サイズ	種類
shot172000_1_bundle.nc	2026年2月17日 13:04	1.9 MB	書類
shot173000_1_bundle.nc	2026年2月17日 13:05	4.3 MB	書類
shot174000_1_bundle.nc	2026年2月17日 13:05	2 MB	書類

異常値フィルタリングの詳細まとめ

- 電子温度 T_e
 - $T_e \leq 0$
 - $T_e \geq 50 \text{ keV}$
- 温度誤差 dT_e
 - $dT_e \leq 0$
 - $dT_e \geq 49 \text{ keV}$
- 電子密度 n_e
 - $n_e \leq 0$
 - $n_e \geq 1 \times 10^{21} \text{ m}^{-3}$
- 密度誤差 dn_e
 - $dn_e \leq 0$
 - $dn_e \geq 1 \times 10^{21} \text{ m}^{-3}$
- EGプレースホルダ値
 - $n_e, dn_e \geq 9 \times 10^5$ (例: 999999)
- $dT_e / T_e \leq 1$
- $dn_e / n_e \leq 1$

→以上の条件に該当する点を NaN に置き換えた。

→後述の可視化は上記フィルタリング後のデータに対する結果を示す。なお、フィルタリング条件は後で柔軟に変更しやすいようにコード設計している。

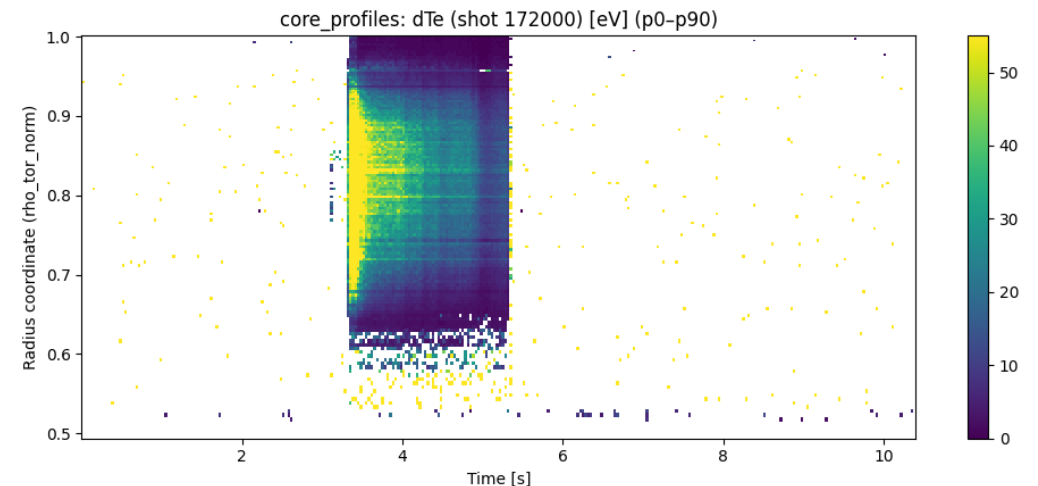
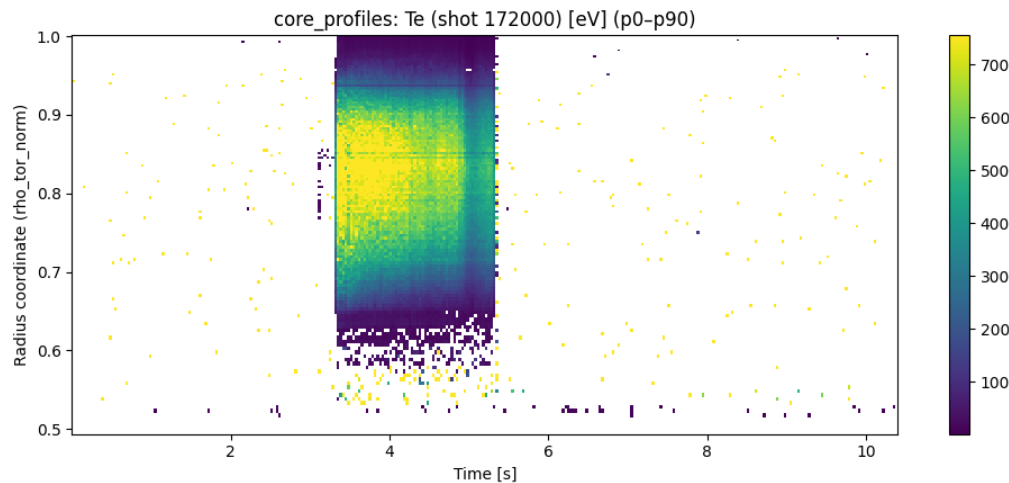
IDSに変換したデータの可視化

・今回使用した IDS に対応した可視化コードを、matplotlib を用いて開発した。

本コードは IDS を入出力 (I/O) として設計されている。→ 装置固有のデータ形式に依存しない可視化層を構築

・ショット番号=172000, 電子温度の可視化例

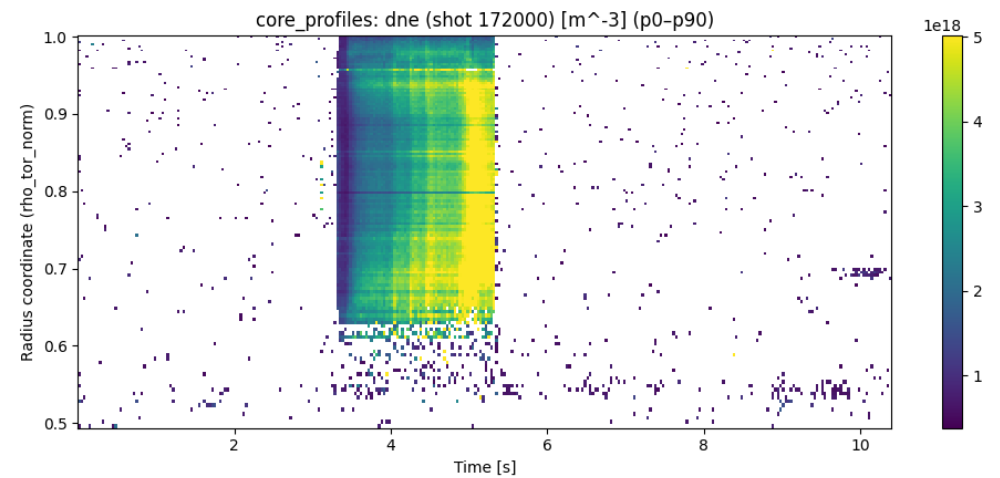
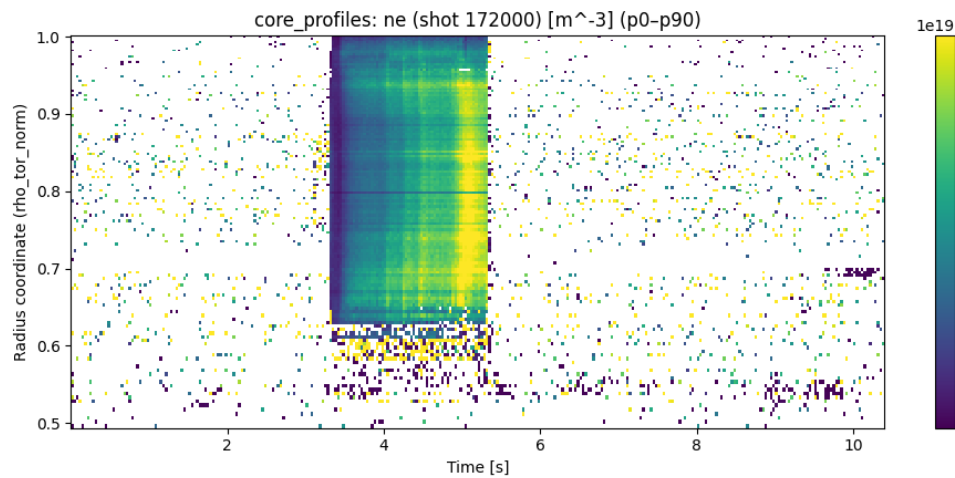
- core_profiles(DimSize =312,139)
- 横軸:時間(s)
 - time range: 0.016953 → 10.38356(312個)
- 縦軸:半径(-)
 - [0.4961, 0.5014, 0.5068, ..., 0.9951, 0.9975, 1.] (139個)
- 左ヒートマップ:電子温度Te(eV)
- 右ヒートマップ:温度誤差dTe(eV)



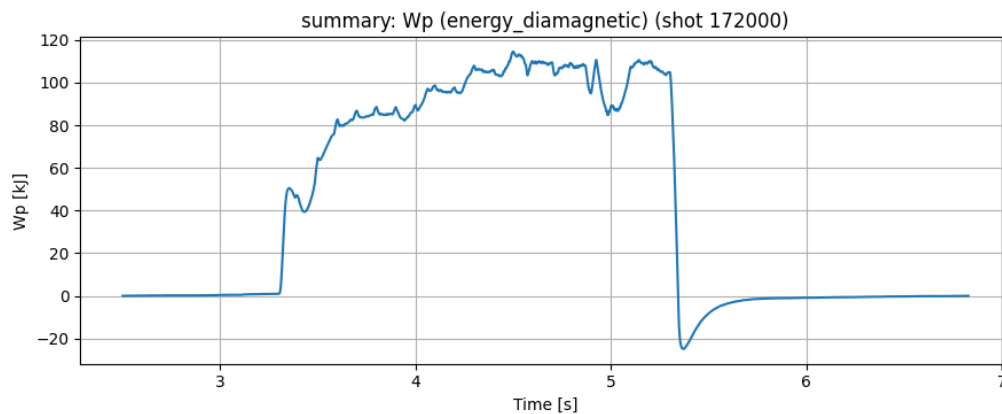
IDSに変換したデータの可視化

• ショット番号=172000, 電子密度の可視化例

- core_profiles(DimSize =312,139)
- 横軸:時間(s)
 - time range: 0.016953 → 10.38356(312個)
- 縦軸:半径(-)
 - [0.4961, 0.5014, 0.5068, ..., 0.9951, 0.9975, 1.] (139個)
- 左ヒートマップ:電子密度 $n_e(m^{-3})$
- 右ヒートマップ:密度誤差 $dne(m^{-3})$



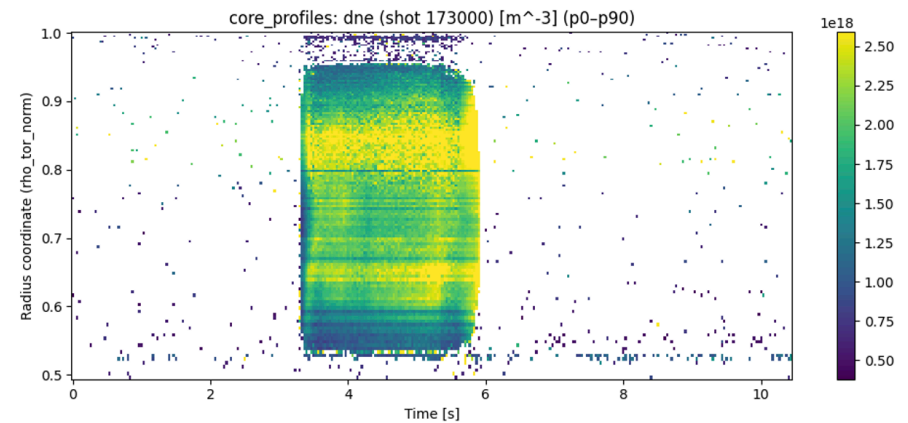
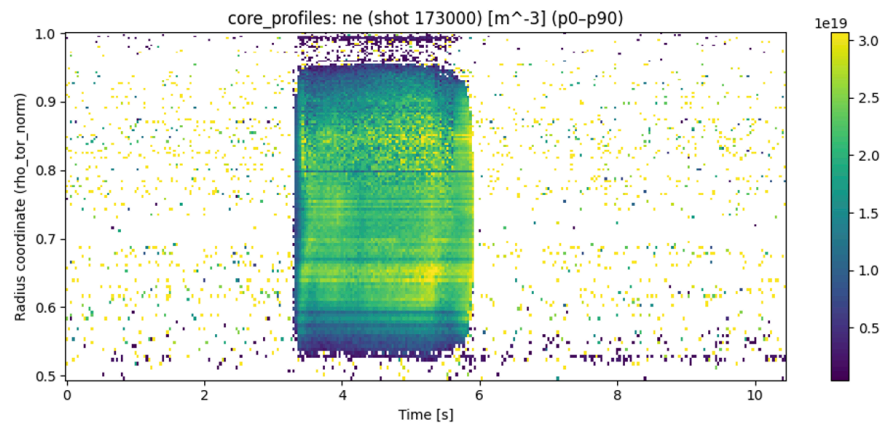
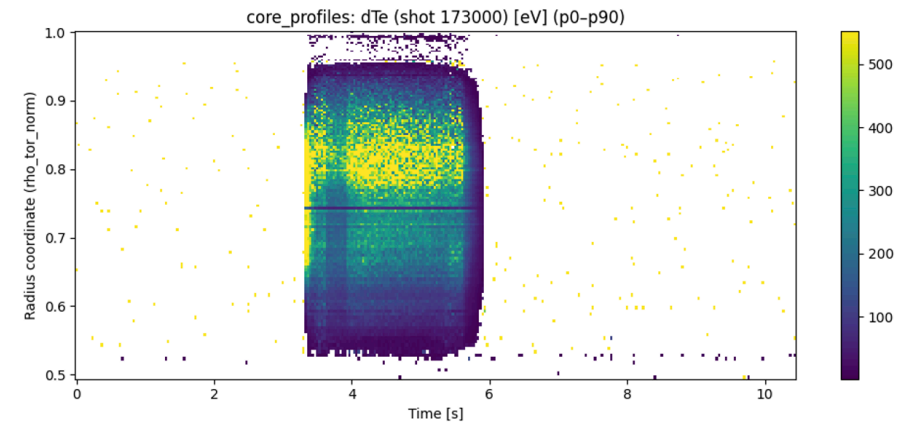
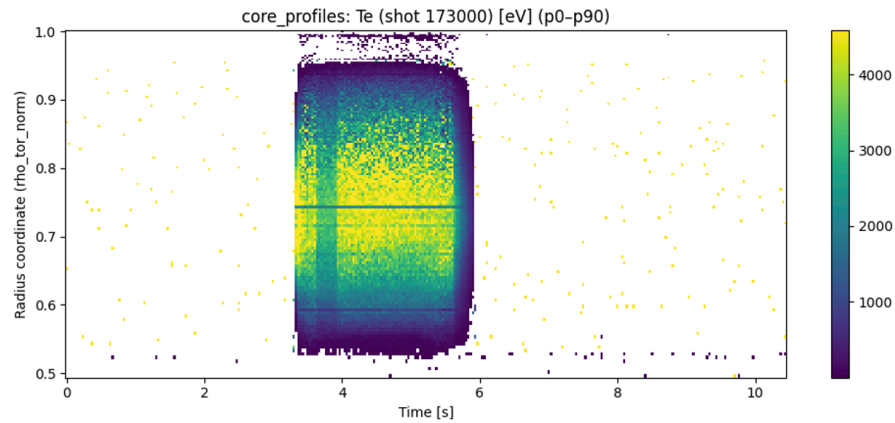
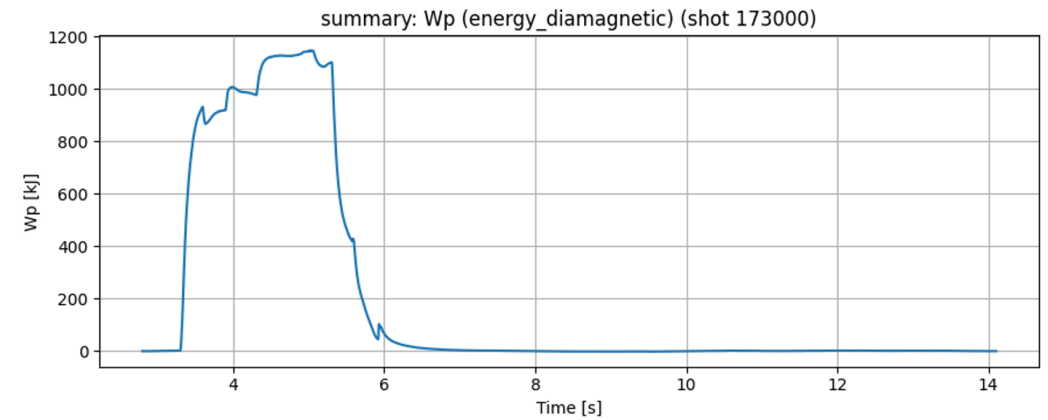
• ショット番号=172000, Wpの可視化例



- summary
- 横軸:時間(s)
 - Time range: 0.0 → 6.83105
 - Start-Time of Integration [sec] = 2.50000
 - Integration time[sec] = 6.831000
- 縦軸:Wp(kJ)
- 見やすいため、単位はIMASのJからkJ変換

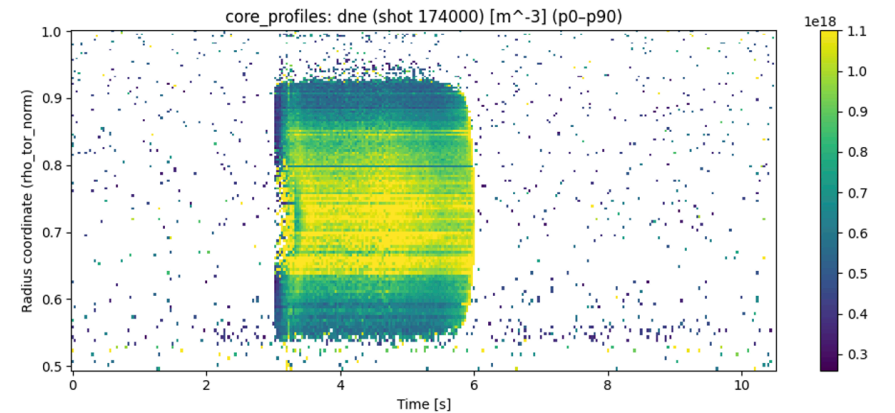
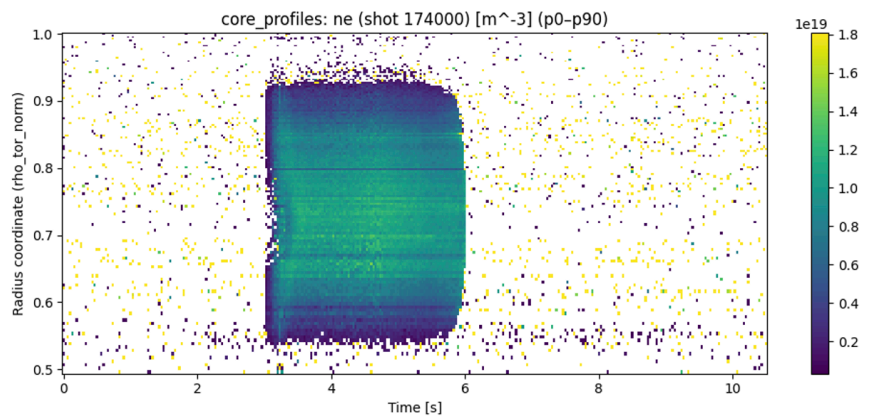
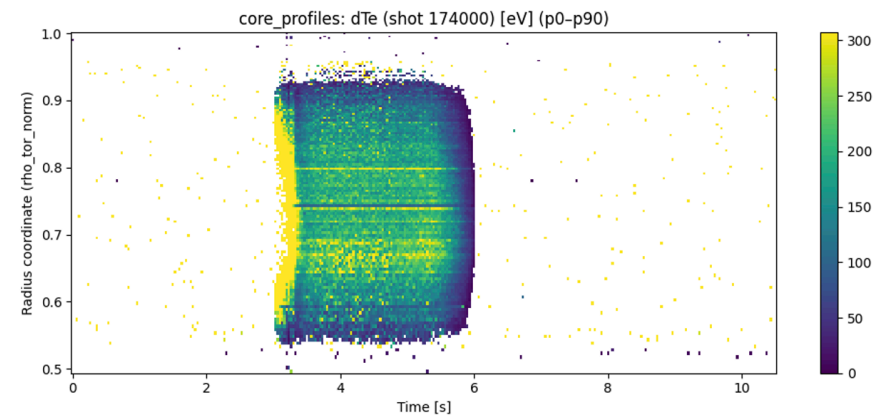
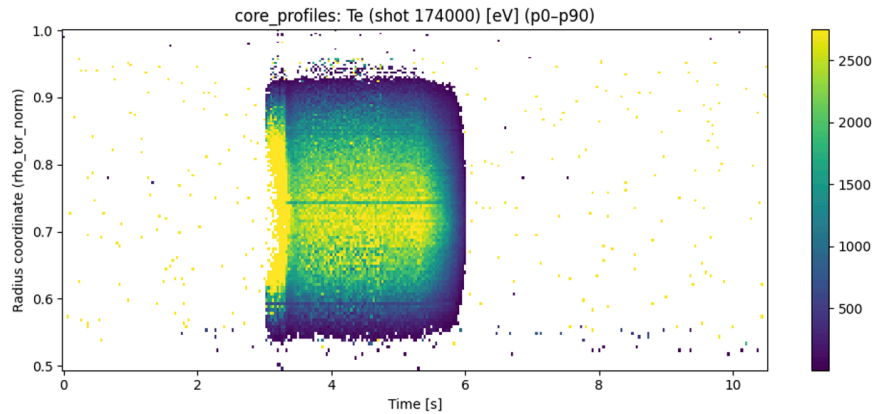
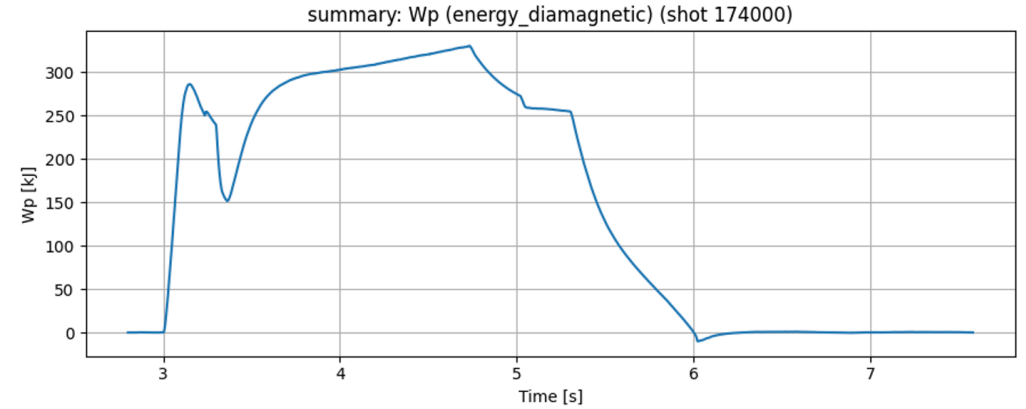
IDSに変換したデータの可視化

- コードの汎用性を確認するため、shot番号173000可視化



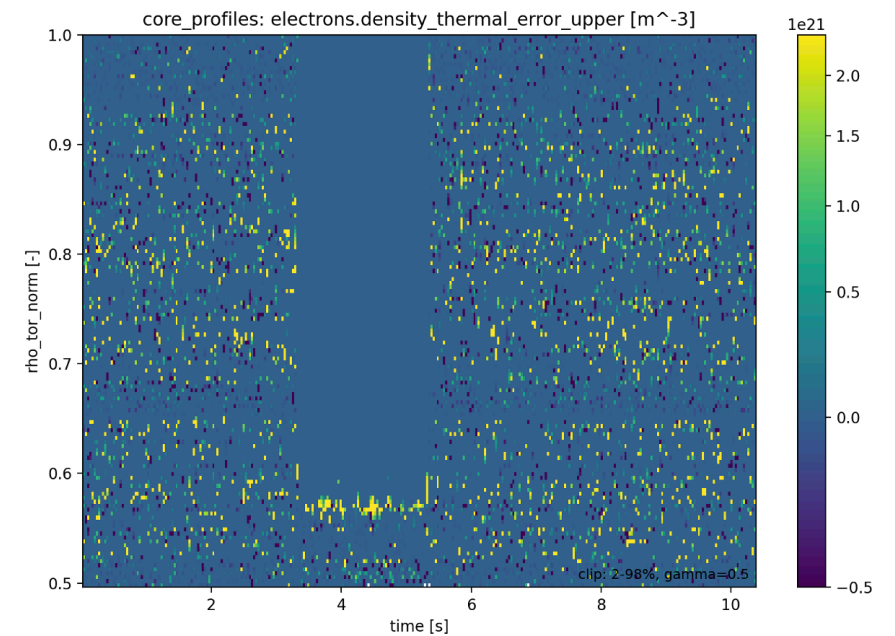
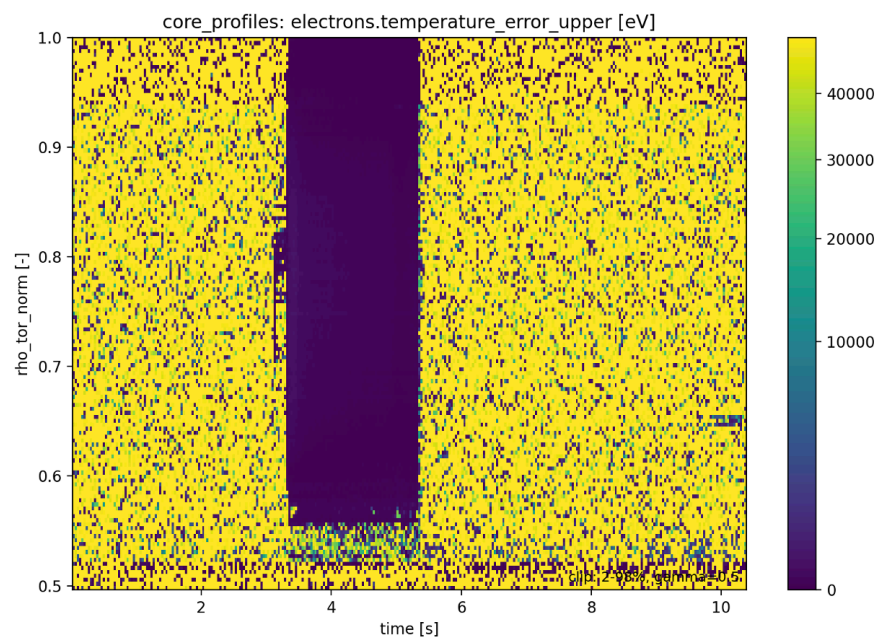
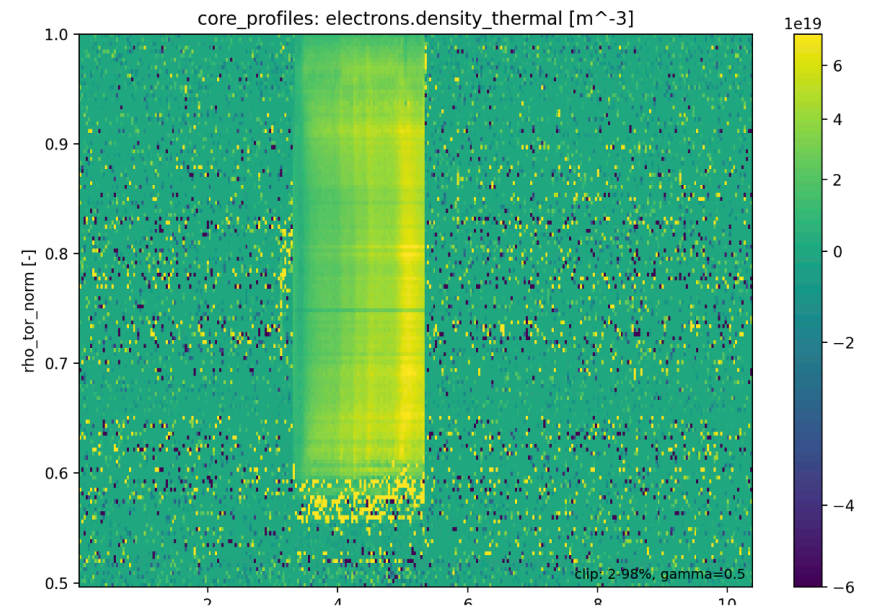
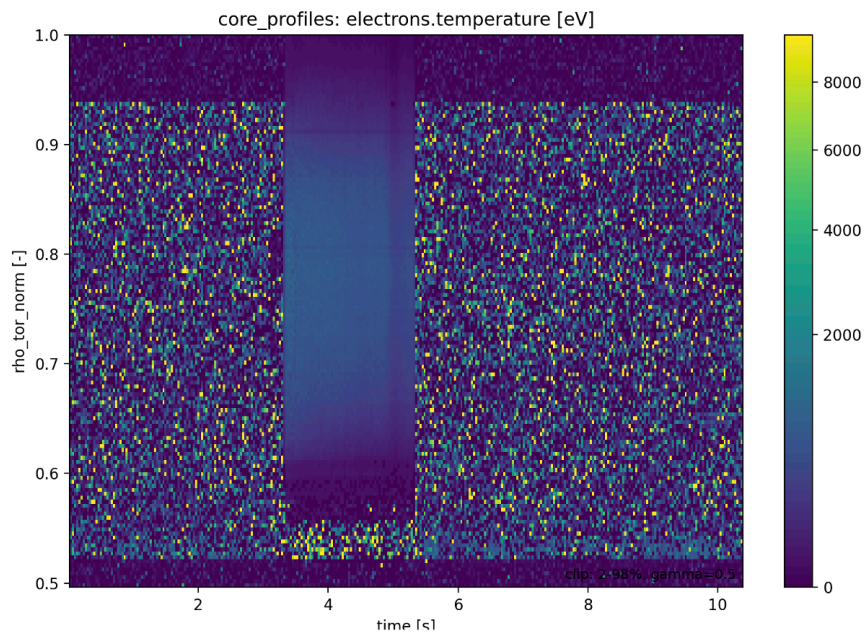
IDSに変換したデータの可視化

- コードの汎用性を確認するため、shot番号174000可視化



IDSに変換したデータの可視化

異常値フィルタリングを行わない場合の可視化例 (shot番号=172000)



LHD解析データ→IDSへの変換コードの流用

以上のように、LHD解析データリポジトリに登録されているEG形式テキストファイルから、IMAS-IDSへの変換・保存コードを作ることができる。CHDデータ解析システムに向けて作成予定の「IMAS-IDS変換・保存用外部アダプターコード群」には、これらのコードの大部分を流用できると想定している。特に、対応表のデザインや外れ値・欠損値の扱いはそのまま流用できるはずである。よって、確認が必要な部分は以下となる。

- ・ CHDデータ解析システム上において、IDS変換アダプターに入る前の診断データがLHD解析データリポジトリにおけるEG形式テキストファイルと比べてどこが異なるか？

→フォーマットのみの問題であれば、読み書き部分（I/O層）と内部表現上での翻訳部分（パース層）にコードを分ければ問題なく対応できる。追加の物理量がある、サンプリングや時間基準が異なる、診断名・チャンネル定義に変更がある場合は、それに応じて対応表のデザイン部分を少し修正する必要がある（+対応するコードの修正）。

まとめと今後の展開

- IMASの調査を行い、NIFS内における活用案を考案した。
- LHD解析データリポジトリの最小限の診断名セット (shotinfo, wp, thomson) に対する、EG形式→IDSの変換・保存コードを試作した。また、IDSをI/Oとする簡単な可視化コードを試作・実証した。
- 今後、対応できる診断名を増やしていく予定である。対応表の作成が最も重要であり、そのデザインには原則としてNIFS内での合意が必要であり、その体制作りを考える必要がある。
 - 診断名・診断名ごとの観測の種類は膨大であり、具体的なコードがない状態で合議によるデザインの決定→その後コード作成は時間がかかりすぎる可能性が高いので、個人的には今回のように少数グループでまず試作→査読のような形でできたものに対する修正コメント→修正し完成版とするような流れが良いのではと考える。
 - 修正は、ほとんどの場合格納するノードの変更や数値処理の微修正だと想定されるので、試作したコード資産は無駄にはならないと考える。
- CHDデータ解析システムにおけるデータI/O形式がどうなるかまだ未決定だと理解しているので (EG形式+netcdf?, 各診断名の書き方・観測はLHDと同じか?)、今回のようにLHD解析データリポジトリのデータを対象としてIDS変換コードを先に作成する活動を継続する予定である (その後CHDに流用)。
- なお、他研究グループにおいて既存のIDS対応表が存在しないかについては、並行して調査を行う必要がある。